REMARKS

Claims 1-21 are pending in the above referenced patent application. Claims 1, 15 and 17 are independent claims.

The examiner uses Hasegawa to reject claims 1-9, 11-13, 15, 17 and 19-21 as having been anticipated.

Claim 1 recites "directing a branch in execution of an instruction stream based on any specified value being true or false and including a token that specifies the number of instructions in the instruction stream that are after the instruction to execute before performing the branch operation.".

Hasegawa neither describes nor suggests directing a branch in execution of an instruction stream based on *any* specified value being true or false. Hasegawa discloses a branch instruction format that includes a region for storing 21, a region for specifying a branch target address 22 and region for storing the number of instructions to branch point 23. (see FIG. 2) Hasegawa does not disclose, suggest or even mention directing a branch in execution of an instruction stream based on *any* specified value being true or false. In fact, the words "value," "true," and "false" are not even mentioned in Hasegawa. Accordingly, claim 1 is not anticipated by Hasegawa.

Applicant canceled claims 15-16.

Claim 17 recites "executing a branch instruction that causes a branch operation in an instruction stream based on any specified value being true or false." As discussed with reference to claim 1 above, Hasegawa neither describes nor suggests executing a branch instruction that causes a branch operation in an instruction stream based on *any* specified value being true or false. Accordingly, claim 17 is not anticipated by Hasegawa.

The examiner uses Hasegawa and Khim Yeoh to reject claim 10 as having been obvious.

Claim 1 is not obvious in view of Hasegawa and Khim Yeoh. As discussed above, Hasegawa does not teach of suggest directing a branch in execution of an instruction stream based on *any* specified value being true or false.

Khim Yeah discloses an improved register-based I/O microcontroller that does not teach or suggest directing a branch in execution of an instruction stream based on any specified value being true or false. Khim Yeah only tests bits in a single flag register that reflect an ALU status but fails to teach or suggest testing within an instruction:

> A flag register 35 is conventionally connected to receive bits that show the status of the ALU operations, for example a bit to show an overflow in the ALU when an add instruction is executed. Other instructions test these bits and for example perform conditional operations such as a branch after an operation to compare the bytes in registers A and B. (col. 3, lines 12-18)

One skilled in this art would not be motivated to combine Hasegawa and Kim because such a combination merely adds an ALU status indicator to a region for storing, a region for specifying a branch target address and region for storing the number of instructions to branch point. No combination yields a branch instruction that causes a branch operation in an instruction stream based on *any* specified value being true or false. Accordingly, claim 1 is not obvious in view of uses Hasegawa and Khim Yeoh.

Claim 10 depends upon, and adds further limitations to, claim 1. Accordingly, claim 10 is not obvious in view of uses Hasegawa and Khim Yeoh.

The examiner uses Hasegawa and Bruckert to reject claims 14, 16 and 18 as having been obvious.

Claim 1 is not obvious in view of Hasegawa and Bruckert. As discussed above, Hasegawa does not teach of suggest directing a branch in execution of an instruction stream based on *any* specified value being true or false.

Bruckert discloses an instruction prefetch system for conditional branching but fails to make up for the deficiencies of Hasegawa:

> (A)n instruction is headed by an operation code or "OP CODE" which may be contained in one or two consecutive storage locations in the memory, followed by one or more operand specifiers. The number of operand specifiers depends on the nature of the operation specified in the operation code portion of the instruction.
>
> As explained in detail in U.S. Pat. No. 4,236,206, each operand specifier may identify the operand to be processed in one of several addressing modes. In some modes, the operand specifier in the instruction may contain the operand itself. In other addressing modes, the operand may comprise or derive from the contents of a register in the fetch unit which is identified by the operand specifier. Alternatively, the operand may be stored in the memory at a location identified by or derived from the contents of a register identified by the operand specifier. (col. 5, lines 5-22)

One skilled in this art would not be motivated to combine Hasegawa and Bruckert because such a combination merely adds prefetch to a region for storing, a region for specifying a branch target address and region for storing the number of instructions to branch point. Accordingly, claim 1 is not obvious in view of Hasegawa and Bruckert.

Claims 14, 16 and 18 depend upon, and add further limitations to, claim 1. Accordingly, claims 14, 16 and 18 are not obvious in view of Hasegawa and Bruckert.

It is believed that all of the pending claims have been addressed. However, the absence of a reply to a specific rejection, issue or comment does not signify agreement with or concession of that rejection, issue or comment. In addition, because the arguments made above may not be exhaustive, there may be reasons for patentability of any or all pending claims (or other claims) that have not been expressed. Finally, nothing in this paper should be construed as

an intent to concede any issue with regard to any claim, except as specifically stated in this paper, and the amendment of any claim does not necessarily signify concession of unpatentability of the claim prior to its amendment.

Please apply any charges or credits to deposit account 06-1050.

Respectfully submitted,

Date: February 17, 2005

Kenneth F. Kozik
Reg. No. 36,572

Fish & Richardson P.C.
225 Franklin Street
Boston, MA 02110-2804
Telephone: (617) 542-5070
Facsimile: (617) 542-8906

21030493.doc